

COUR 2

LA BLOCKCHAIN : PROTOCOLE ET ARCHITECTURE



PARTIE I

-

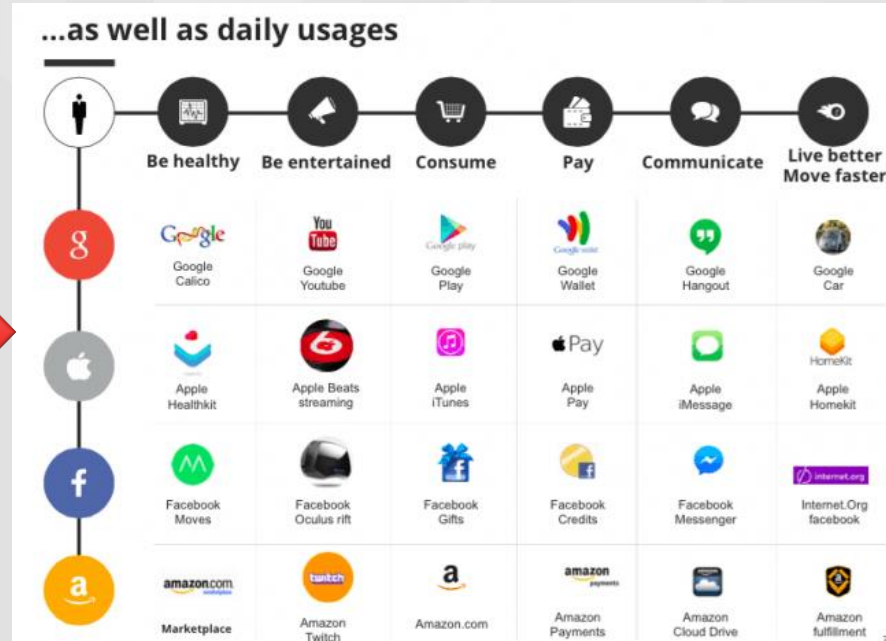
BLOCKCHAIN ARCHITECTURE ET PROTOCOLE

- Centralisation de l'information
- Le problème des généraux Byzantins
- Les algorithmes cryptographiques*
- Le chainage des transactions
 - La topologie réseau
- Structure des données du registre
 - Transaction UTXO
- Hard Fork & Soft Fork

Les promesses d'internet et d'une communication décentralisée :

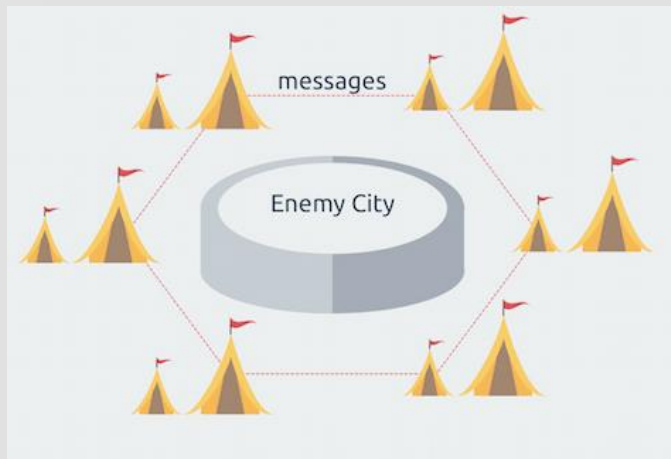
- Un réseau informatique partagé (ARPANET – 1961 & CYCLADES – 1970)
- Un protocole de communication publique (TCP/IP – 1983 & WWW – 1990)
- La neutralité du réseau en principe fondateur
- Capacité de diffusion de l'information dans le monde entier

Où en est-on aujourd'hui ?



Problématique d'une armée Byzantine assiégeant une ville

1. Camps militaires séparés géographiquement (décentralisés)
2. Communiquer afin de mettre au point une stratégie commune d'attaque ou de retrait. On parle d'atteindre un consensus.
3. Existence probable de traîtres qui peuvent falsifier ou ne pas délivrer un message



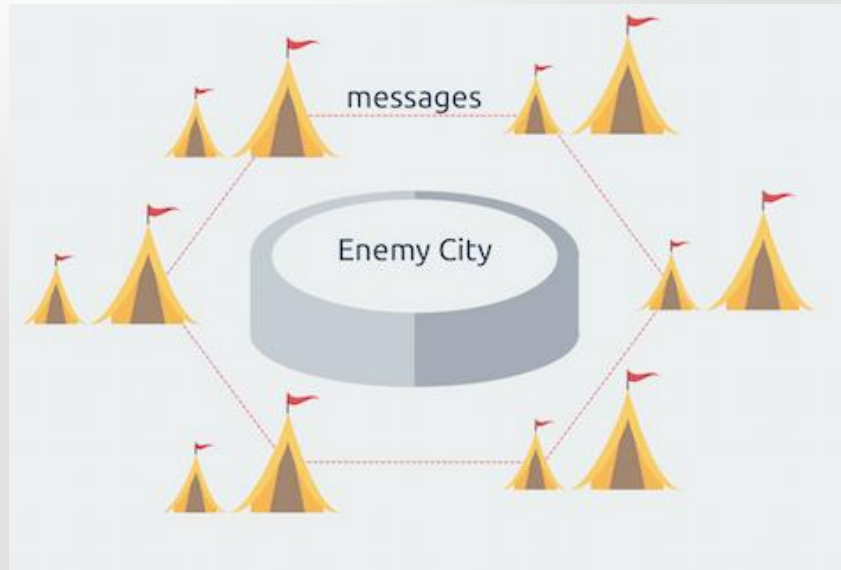
Le problème des généraux Byzantins



Les risques d'une défaite :

- Tout défaut présentant des informations erronées ou incohérentes aux différents acteurs qu'on résume par le terme **Byzantine fault tolerance**
- La perte de confiance du système du fait de **"Byzantine fault tolerance"** conduisant à un échec de l'ensemble des acteurs a **"Byzantine failure"**



La naissance d'un protocole de communication sans tiers de confiance !



	The Situation	
Agree on a Strategy	Objective	Agree on Valid Transactions
Separated Camps	Spacial Distribution	Distributed Nodes in the Network
Loyal Troop and Loyal Generals	The Good Ones	Truthful Nodes
Traitors	The Bad Ones	Evil Nodes
Corrupt a Message	The Attack	Add an Invalid Transaction to the Blockchain
How to Know which Message is True	The Problem	How to know which Transaction is Valid
Don't Have	A Solution	Proof of Work
Don't Have	Consensus	Blockchain with More Combined Difficulty

Les principaux algorithmes cryptographiques de consensus :

1. La preuve de travail « Proof of Work (PoW) » utilisée notamment par Bitcoin
2. La preuve de possession « Proof of Stake (PoS) » utilisée par NXT

Le système de preuve de travail ou de possession se base systématiquement sur une fonction de hachage cryptographique (SHA256, Script, Blake etc...).

Quel sont leurs rôles ?

- Etablir les règles de consensus à savoir de constitution, de chaînage, de validation et de propagation de Bloc de transaction(s) au sein du réseau pair à pair par l'utilisation de fonction cryptographique.
- Protection contre des attaques de type (Double Spending, Sybil, DoS ...)

La preuve de travail « Proof of Work (PoW) » sur un réseau distribué :

Un processus computationnel requérant de la puissance CPU afin de résoudre un problème mathématique complexe dont le résultat est simple à vérifier.

Dans le cas de Bitcoin on utilise le système de preuve de travail Hashcash* qui permet la génération et la validation de Bloc.

Un Bloc **B** est accepté et ajouté à la chaîne de Bloc existante si la valeur de son hash(B) commence par « 0000xxxxxxxx » du fait que : **hash(B) ≤ M/D**

M : entier non signé indique le seuil cible « Target nBits »

D représente la difficulté « Difficulty » de la tâche

Il existe deux catégories de paramètres de l'algorithme de consensus PoW :

Consensus paramètres	Objectif	
nSubsidyHalvingInterval	Diminuer la récompense "rewards" des mineurs à "n" bloc (bitcoin : 210000 // 4 ans)	
nPowTargetTimespan	Analyse periodique du seuil de difficulté en seconde (bitcoin : $14 * 24 * 60 * 60$; // 2 semaines)	
nPowTargetSpacing	Intervalle de temps entre chaque Bloc en seconde (bitcoin : $10 * 60$; // 10 min)	
DifficultyAdjustmentInterval	Réajustement de la difficulté à valider un Bloc (bitcoin : $nPowTargetTimespan / nPowTargetSpacing = 2016$ Bloc // 14 jours)	
Bloc paramètres	Objectif	Mise à jour lorsque
Version	Version logiciel appliquée au Bloc	Une nouvelle version applicative est active
hashPrevBlock	256-bit hash du précédent block header	Un nouveau Bloc est en cours de validation
hashMerkleRoot	256-bit hash de l'ensemble des transactions du Bloc	Une transaction est accepté
Time	Timestamp en seconde depuis 1970-01-01T00:00 UTC	Chaque seconde passant
Bits	Seuil en format compact	la difficulté est réajusté
Nonce	32-bit nombre (départ de 0)	Un hash est généré (incrémentacion)

Un exemple de fonctionnement de l'algorithme PoW :

```
<?
//This reverses and then swaps every other char
function SwapOrder($in){
    $Split = str_split(strrev($in));
    $x='';
    for ($i = 0; $i < count($Split); $i+=2) {
        $x .= $Split[$i+1].$Split[$i];
    }
    return $x;
}

//makes the littleEndian
function littleEndian($value){
    return implode (unpack('H*',pack("V*", $value)));
}

$version = littleEndian(1);
$prevBlockHash = SwapOrder('00000000000008a3a41b85b8b29ad444def299fee21793cd8b9e567eab02cd81');
$rootHash = SwapOrder('2b12fc1b09288fcaff797d71e950e71ae42b91e8bdb2304758dfcfc2b620e3');
$time = littleEndian(1305998791);
$bits = littleEndian(440711666);
$nonce = littleEndian(2504433986);

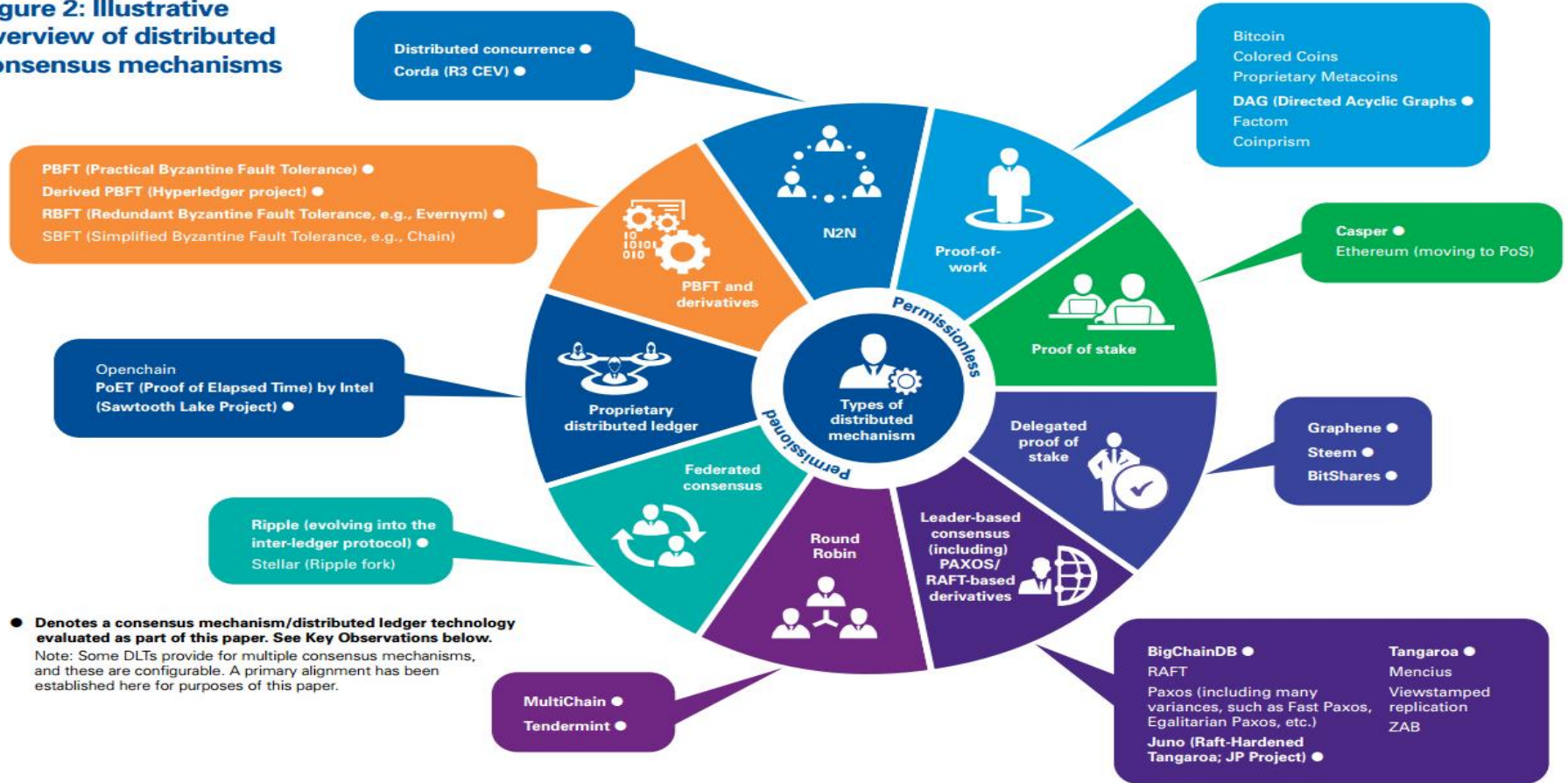
//concat it all
$header_hex = $version . $prevBlockHash . $rootHash . $time . $bits . $nonce;

//convert from hex to binary
$header_bin = hex2bin($header_hex);
//hash it then convert from hex to binary
$pass1 = hex2bin( hash('sha256', $header_bin ) );
//Hash it for the second time
$pass2 = hash('sha256', $pass1);
//fix the order
$FinalHash = SwapOrder($pass2);

echo $FinalHash;
?>
```

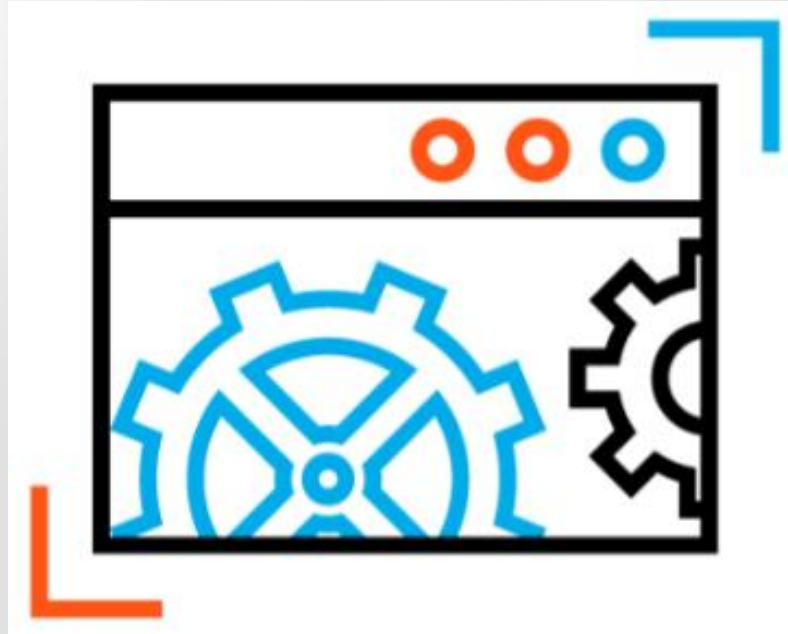
Les algorithmes de consensus cryptographique

Figure 2: Illustrative overview of distributed consensus mechanisms



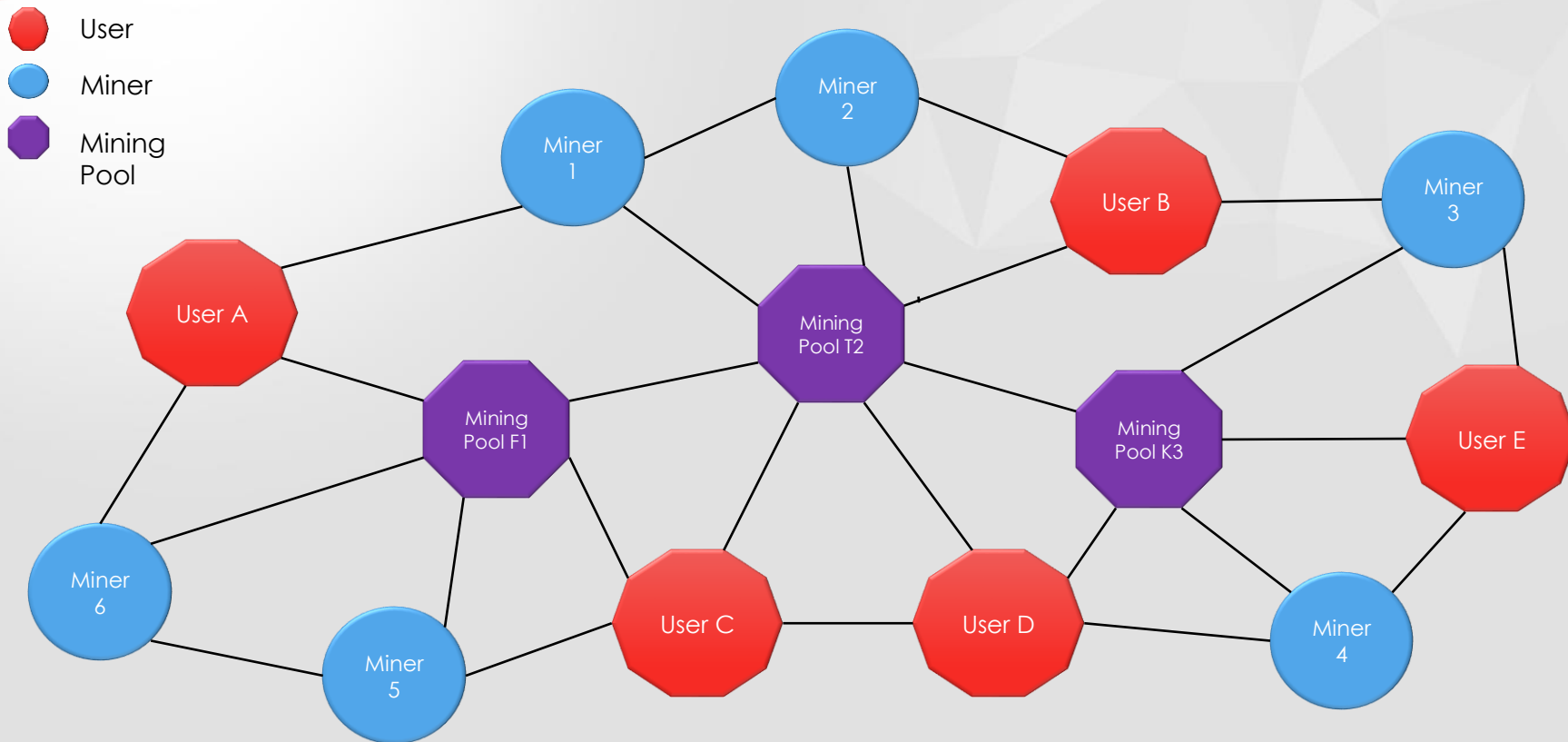
BLOCKCHAIN ARCHITECTURE ET PROTOCOLE

Démonstration



La topologie du réseau

Les acteurs du réseau P2P :

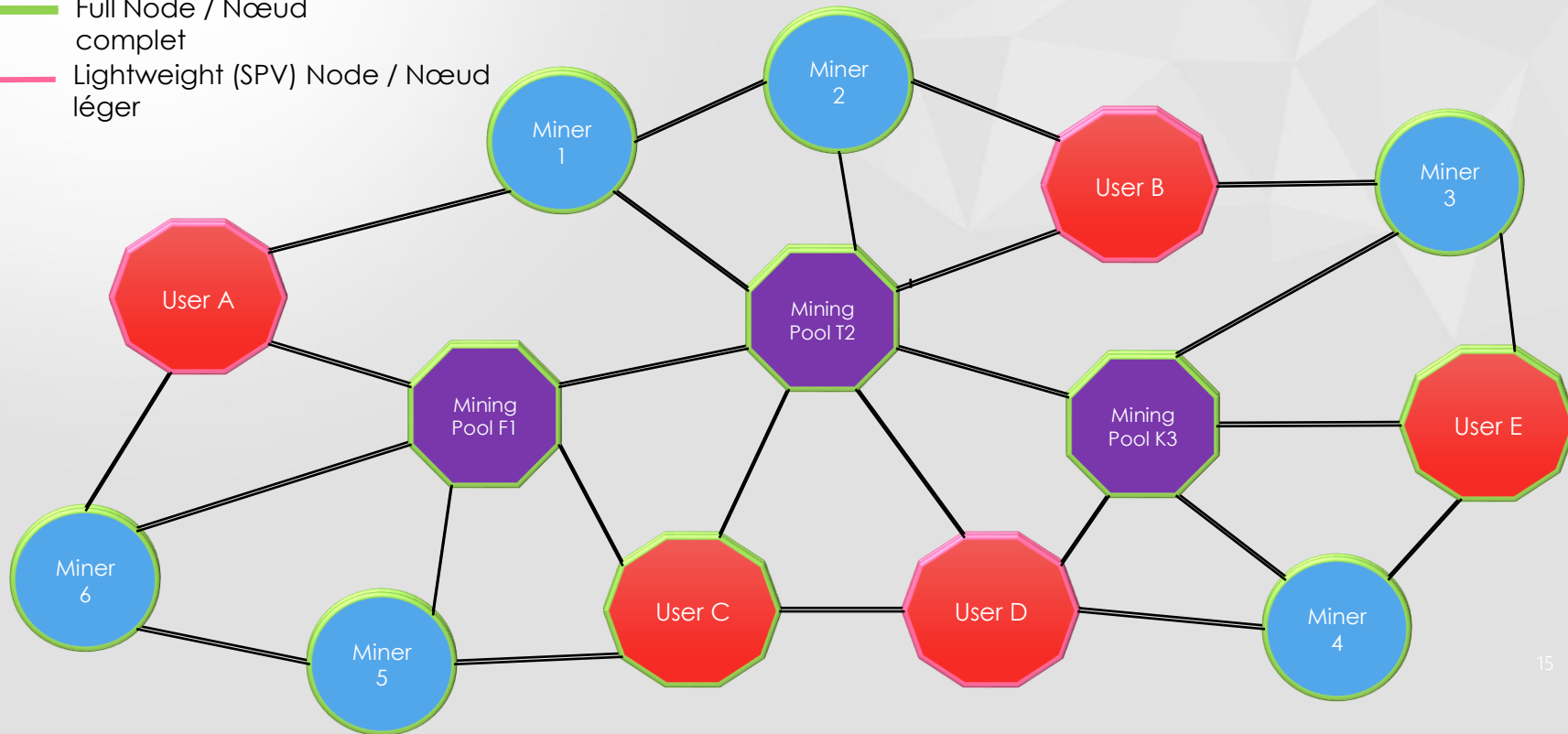


La topologie du réseau

Le rôle des acteurs du réseau P2P :

— Full Node / Nœud complet

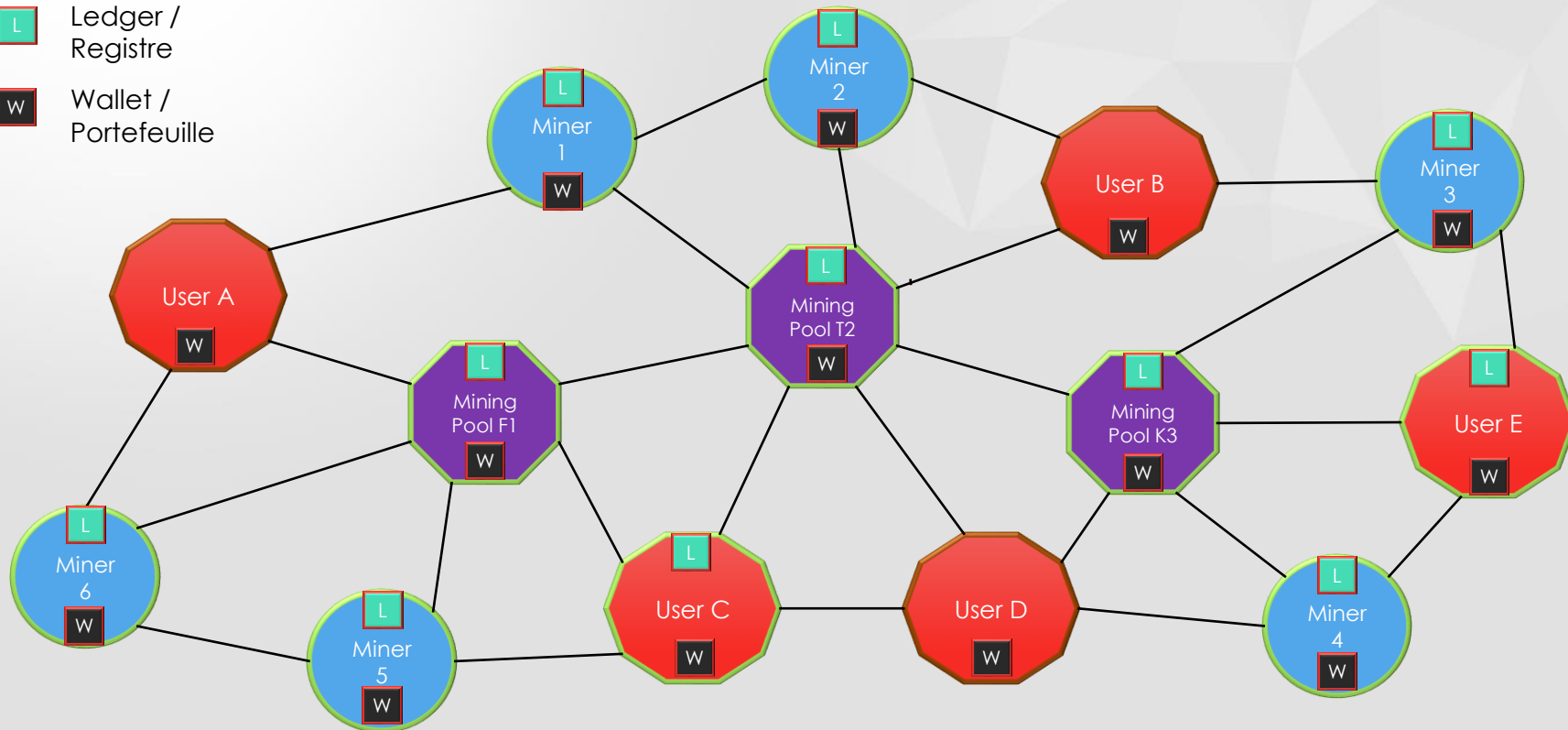
— Lightweight (SPV) Node / Nœud léger



La topologie du réseau

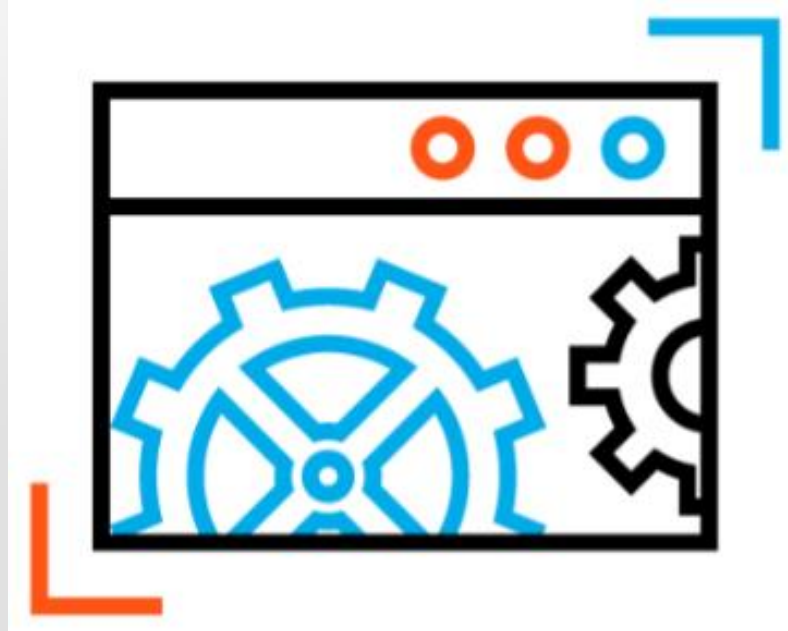
Les principales fonctions des acteurs du réseau P2P :

-  Ledger /
Registre
-  Wallet /
Portefeuille



BLOCKCHAIN ARCHITECTURE ET PROTOCOLE

Démonstration



Quelques derniers chiffres du réseau Bitcoin :

- Nombre journalier de transaction : 350.000 Trans/j (source : blockchain.info)
- Taille moyen d'un Bloc : 1 Megabytes
- Taille totale actuel de la Blockchain Bitcoin : 150 Gigabytes
- Vitesse de propagation d'un Bloc reçu par 50% des nœuds : 1.75 sec (source : bitcoinstats.com)
- Vitesse de propagation d'une transaction reçu par 50% des nœuds : 3.6 sec (source : bitcoinstats.com)

L'arbre de Merkle ou arbre de hachage garant du registre distribué :

Cette structure de donnée inventée par Ralph Merkle en 1979 est utilisée pour résumer toutes les transactions dans un bloc, produisant une empreinte numérique globale de l'ensemble des transactions, fournissant un processus très efficace pour vérifier si une transaction est incluse dans un bloc.

Comment est il construit ?

Un arbre de Merkle est construit par des paires de noeuds hachées récursivement jusqu'à ce qu'il n'y ait qu'un seul hash, appelé root ou merkle root.

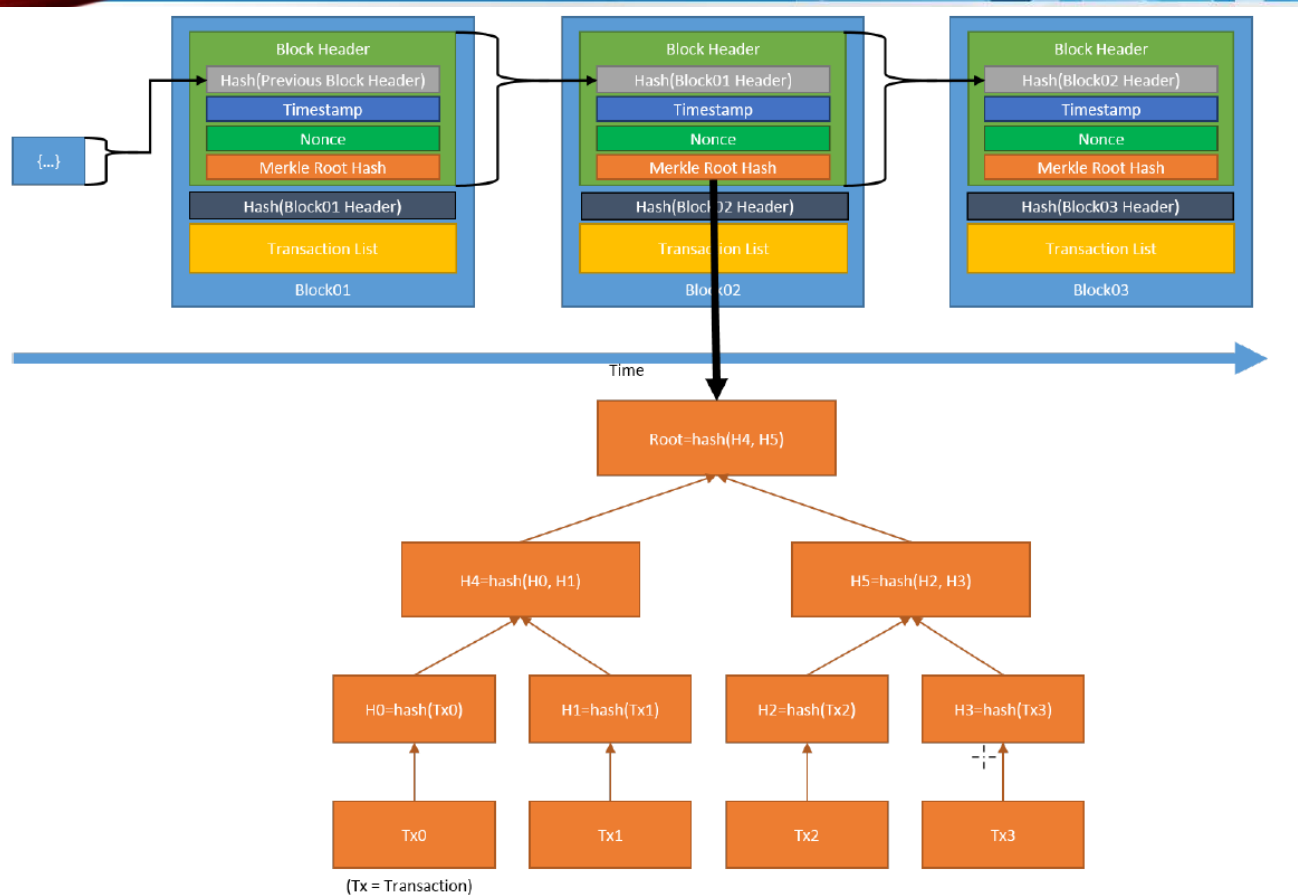
L'algorithme de hachage cryptographique utilisé dans les arbres Merkle de bitcoin est SHA256 appliqué deux fois, également appelé double-SHA256.

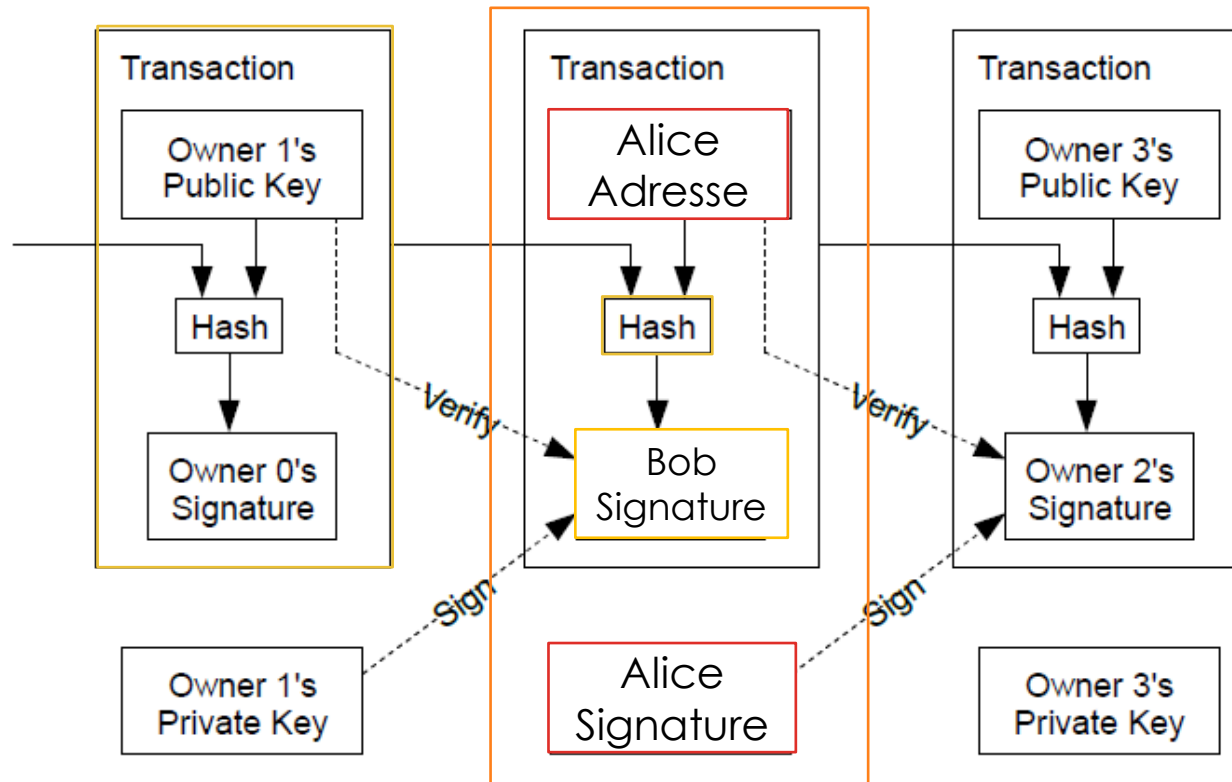
Volumétrie de l'arbre de Merkle :

L'efficacité des arbres Merkle devient évidente à mesure que l'échelle augmente. Le tableau montre la quantité de données qui doivent être échangées en tant que chemin Merkle pour prouver qu'une transaction fait partie d'un bloc.

Number of transactions	Approx. size of block	Path size (hashes)	Path size (bytes)
16 transactions	4 kilobytes	4 hashes	128 bytes
512 transactions	128 kilobytes	9 hashes	288 bytes
2048 transactions	512 kilobytes	11 hashes	352 bytes
65,535 transactions	16 megabytes	16 hashes	512 bytes

Structure des données du registre





Transaction UTXO

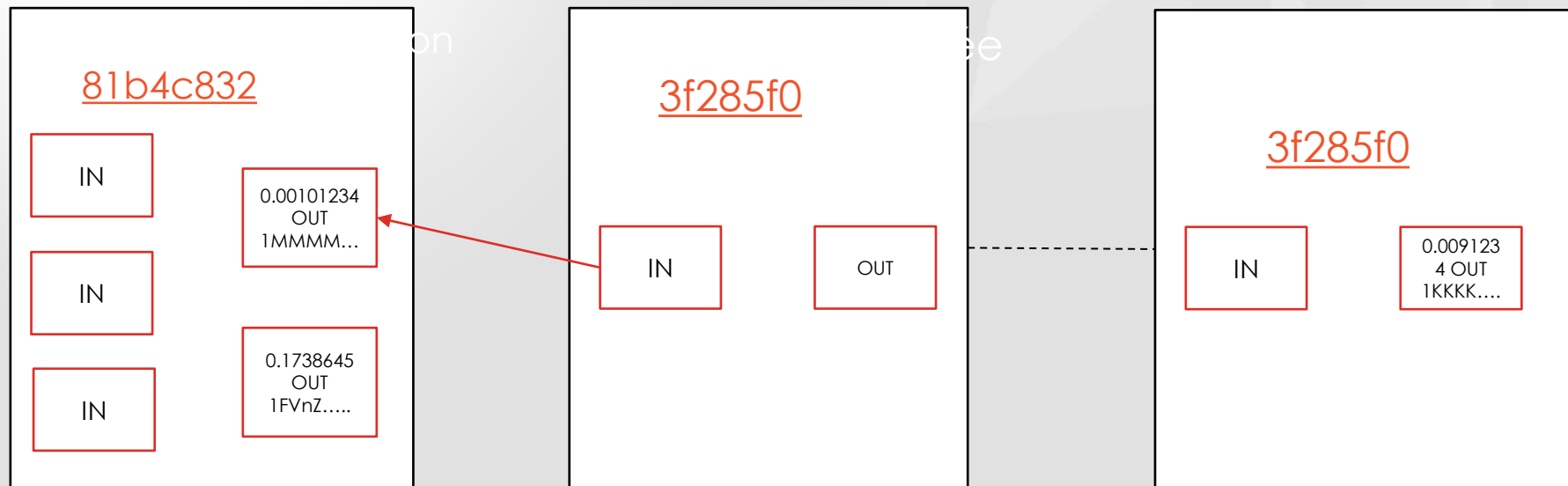
Transaction UTXO (Unspent Transaction Output) :

La construction fondamentale d'une transaction bitcoin correspond à une sortie de transaction non dépensée (UTXO). UTXO sont des unités de bitcoin verrouillées à un propriétaire spécifique, enregistrés sur la chaîne de blocs et reconnus comme unités par l'ensemble du réseau.

Aucun soldes ou comptes dans Bitcoin mais un éparpillement d'UTXO enregistrés dans la Blockchain

Le réseau bitcoin suit tous les UTXO disponibles (non utilisés) actuellement en plusieurs millions. Chaque fois qu'un utilisateur reçoit une unité de bitcoin, ce montant est enregistré dans la Blockchain comme étant un UTXO.

Création d'une série de transaction UTXO



Transaction

Afficher les informations d'une transaction bitcoin

3f285f083de7c0acabd9f106a43ec42687ab0bebe2e6f0d529db696794540fea

1MMMSUB1piy2ufrSguNUdFmAcvqrQF8M5



1KKKK6N21XKo48zWKuQKXdvSsCf95ibHFa

0.00091234 BTC

0.00091234 BTC

Récapitulatif

Taille	223 (octets)
Date de réception	2014-01-07 06:24:53
Inclue dans les blocs	279068 (2014-01-07 06:22:00 + -3 minutes)
confirmations	186200 confirmations
Relayée par l'IP	62.65.111.110 (whois)
Visualiser	Voir le graphique

Entrées et sorties

Total des entrées	0.00101234 BTC
Total des sorties	0.00091234 BTC
Taxes	0.0001 BTC
Fee par octet	44.843 sat/B
Estimation des BTC échangées	0.00091234 BTC
scripts	Voir les scripts et coinbase

Transaction

Afficher les informations d'une transaction bitcoin

81b4c832d70cb56ff957589752eb4125a4cab78a25a8fc52d6a09e5bd4404d48

1MvJZhYsoF1AD1h1uXTR15d1ZhJMHGAJR8

1MWvGVgCmgBqxx1iJt77YJyeSfTjX6JnAC

1MaGwACpvR9AjDFvW9UtJshQHhZNSa6MtS



1MMMSUbb1piy2ufrSguNUdFmAcvqrQF8M5

1FVnZDLN2c2RqnqLkySoeYD9n7BiYdPhMv

0.00101234 BTC

0.01738645 BTC

0.01839879 BTC

Récapitulatif

Taille 617 (octets)

Date de réception 2014-01-05 07:00:06

Inclue dans les blocs [278696](#) (2014-01-05 07:00:58 + 1 minutes)

confirmations 186434 confirmations

Relayée par l'IP [Blockchain.info](#)Visualiser [Voir le graphique](#)

Entrées et sorties

Total des entrées 0.01859879 BTC

Total des sorties 0.01839879 BTC

Taxes 0.0002 BTC

Fee par octet 32.415 sat/B



Estimation des BTC échangées 0.01738645 BTC

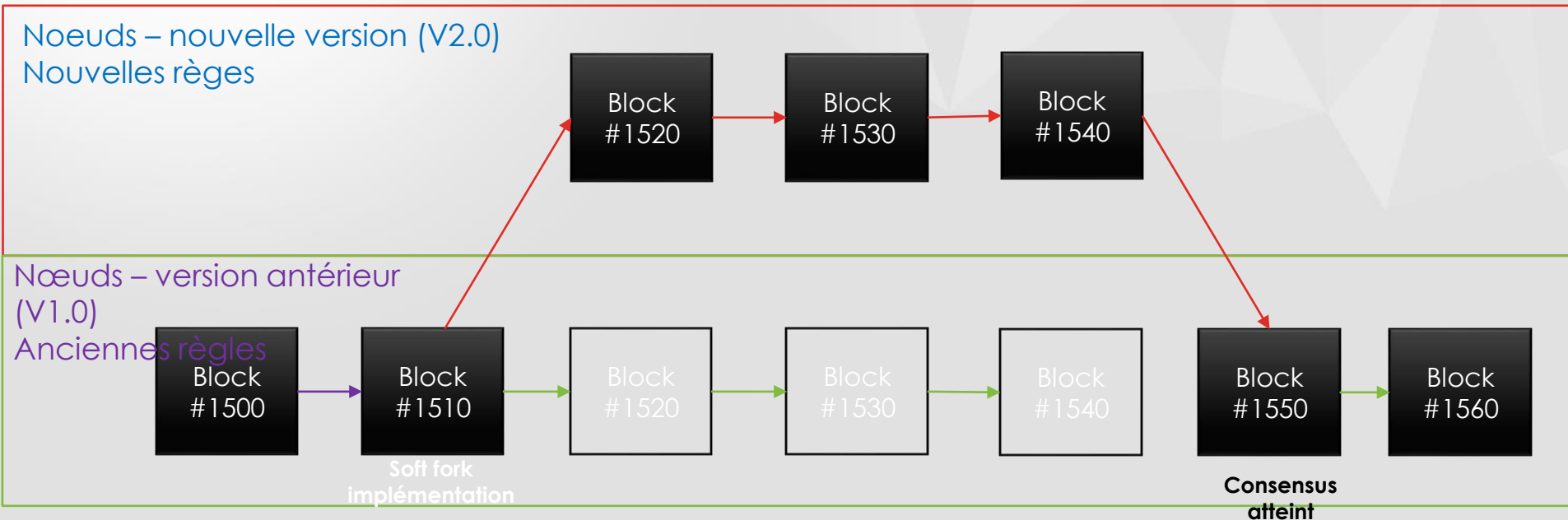
scripts [Voir les scripts et coinbase](#)

BLOCKCHAIN PROTOCOLE

Hard Fork & Soft Fork

Soft fork explication :

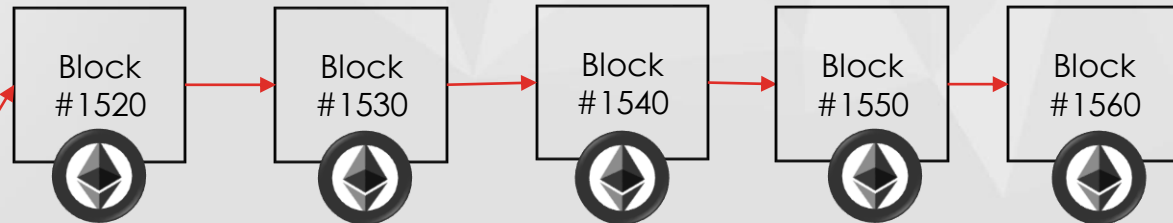
-  Chaîne de blocs la plus longue
-  Blocs violant les nouvelles règles



Hard fork explication :

-  Chaîne de blocs d'origine
-  Nouvelle chaîne de Blocs

Noeuds – nouvelle version (V2.0)
Nouvelles règles



Nœuds – version antérieure
(V1.0)

**Hard fork
implémentation**

Anciennes règles

